



InterActive 4 Dumb Terminal

Written By: Steven Robert Cypherd



PARTS:

- [Basic Stamp 2 \(1\)](#)

SUMMARY

Here we are making a dumb terminal out of an old DMC 16117A 1x16 LCD display. The first thing to do is to get a serial interface for the display. I am using a [SparkFun SerLCD](#) for this. The 16117A only has 14 connectors and that is fine. The 15 & 16 connections on the SerLCD are for the back light LEDs. Just carefully cut the header and remove those connectors. Install the SerLCD on to the 16117A. Connect Power, Ground and Rx to your processor port. Serial communications parameters are vanilla 9600 baud, 8 bits even or true, 1 Start bit, 1 Stop bit, no parity.

I am just using fragments of my code for this article. It is the same code I used for the Serial EEPROM article. I just changed the output drivers to handle the 16117A display.

The code text is on [Let's Make Robots](#) and [Instructables](#).

If you want back lights you will have to remove the mirror backing on the 16117A. On mine the foil was on another piece of plastic. I removed all of it. The lights can be anything. Connect them plus voltage to 15 and ground on 16.

I do not remember when I got my 16117A. It was one of the first LCD displays available for processor boards way back in the mid 90's. No one did much with it because of its horrible


tedious parallel interface. The serial interfaces now make it great for a dumb terminal attached to a processor board like the Basic Stamp 2. You still have to baby this display and the processor board helps. Power glitches and static discharge send the 16117A into uncontrollable states that only a correct series of commands can fix. The Parallax BASIC should be easy to translate into other systems.

The 16117A controller is for multi-line displays. The 16 -display is really two 8-byte lines in the display's memory; 0-7 for the first line and 64-71 for the second line. In my program to make the cursor position you add the control character 128 to it. That tells SerLCD that it is a cursor position. The memory is a mess. See the manual.

Having the SerLCD gives you all those pins to do other things with. Run switches for user feedback. Run things for the user interface. Make a bigger interactive thing. The display keeps the users informed. And the 16117A looks almost normal thanks to the other processor board. Just to send a long string to the display you need a processor. Send first 8 letters then set the cursor position and print the rest of the string to the display. That is what my interface does.

Every processor board supports network serial communications on most of their pins. Networks must have the grounds connected or they will not understand each other. Minimum is Rx and Ground. Long network cables should be shielded twisted-pair cables. Pull-up resistors is all you need. About 47K. See my InterActive articles. I have not got the network stuff done yet, so no code file this time. I wish I had friends to share this with. I am a very lonely geek in LA.

Step 1 — InterActive 4 Dumb Terminal

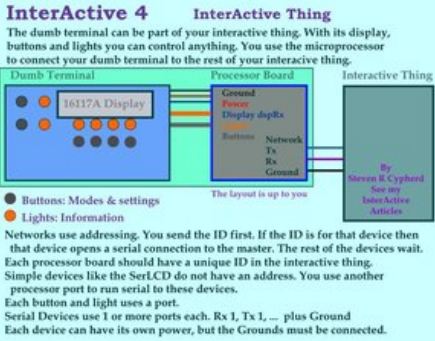


InterActive 4 Dumb Terminal

16 15 D7 D6 D5 D4 E R_w R_s V_d V_s
Back lights G V+ 14 13 12 11 ICSP 6 5 4 3 2 1

DMC 16117A
ALE91964 91351
OPTREX JAPAN

The Serial Interface from SparkFun.com.
Use 9600 baud even or true, 1 Start bit, 1 Stop bit, No Parity. Maybe re-programmed through the ICSP port.
See the manual.



InterActive 4 InterActive Thing

The dumb terminal can be part of your interactive thing. With its display, buttons and lights you can control anything. You use the microprocessor to connect your dumb terminal to the rest of your interactive thing.

Dumb Terminal **Processor Board** **Interactive Thing**

Buttons: Modes & settings
Lights: Information

Networks use addressing. You send the ID first. If the ID is for that device then that device opens a serial connection to the master. The rest of the devices wait. Each processor board should have a unique ID in the interactive thing. Simple devices like the SerLCD do not have an address. You use another processor port to run serial to these devices. Each button and light uses a port. Serial Devices use 1 or more ports each. Rx 1, Tx 1, ... plus Ground. Each device can have its own power, but the Grounds must be connected.

The layout is up to you

By Steven R Cypherd
See my Interactive Articles

InterActive 4 A 16117A FIX

The 16117A has many problems so here is my fix for this 1x16 display. All numbers are in Decimal. Command = Function: Parts

```

...
ELSEIF tmp1 = 80 THEN /Command Set
  DEBUG "Enter string # to display", CR
  DEBOUT DEC tmp5
  IF tmp5 = 30 THEN /Enter the fix
    FOR tmp1 = 1 TO 15
      SEROUT Dsp1, 84, [DspCmd1] /Must be even
      SEROUT Dsp1, 84, [DspPunc] /40 = P32: DL 41, W 2 line, P 5x7
      PAUSE 5 /Number lines 2
      NEXT tmp5 = 0
      SEROUT Dsp1, 84, [DspCmd1] /12 = C8: D on, C off, B off
      PAUSE 5 /Display On
      SEROUT Dsp1, 84, [DspCmd1] /Cursor Off
      SEROUT Dsp1, 84, [DspCmd1] /Link Off
      PAUSE 5
      SEROUT Dsp1, 84, [DspCmd1] /1 = Clear display
      PAUSE 5
      SEROUT Dsp1, 84, [DspCmd1]
      PAUSE 5
      SEROUT Dsp1, 84, ["Done"]
    ENDIF
  GOTO StringsOut
ENDIF
...

```

By Steven R Cypherd

- SparkFun.com Serial LCD interface
- Dumb Terminal
- 16117A FIX

Step 2

InterActive 4 16117A Displaying a Long String

Because of the 16117A controller being for multi-line displays the addressing is a mess. Hello DMC16117A The 1 line 16117A is two 8 byte lines 64 bytes from the first line.

0 - 764 - 71

MyStr is 17 bytes. I move Hi 8 up by one. Insert a zero. Print It. Set cursor to 64. Move Hi 8 to front. Add 0. Print It.

Hello DMQC16117A Print Hello DM

12345678901234567

C16117A0 Print C16117A

123456789

Display Hello DMC16117A

1234567890123456

```

...
IF DspStr = 0 THEN noDspStr
DEBUG DEC tmp1, ". ", STR myStr, CR
SEROUT Dsp1, 84, [DspCmd1]
SEROUT Dsp1, 84, [DspClear]
PAUSE 10
FOR tmp2 = 0 TO 15 Go through string
  IF myStr(tmp2) = 0 AND tmp2 < 8 THEN
    SEROUT Dsp1, 84, [STR myStr]
  EXIT
ENDIF
IF tmp2 = 8 THEN /Start Second line
  tmp4 = 16
  tmp3 = 15
  By Steven R Cypherd ...

```

```

DO
  myStr(tmp4) = myStr(tmp3)
  tmp4 = tmp4 - 1 /Move Hi 8 Up one
  tmp3 = tmp3 - 1
  LOOP UNTIL (tmp3 < 8)
  myStr(tmp2) = 0
  SEROUT Dsp1, 84, [STR myStr]
  PAUSE 10
  tmp3 = 64 + 128 'Set Address to Line 2
  SEROUT Dsp1, 84, [DspCmd1]
  SEROUT Dsp1, 84, [tmp3]
  PAUSE 15
  tmp3 = 0
  tmp4 = tmp2 + 1
  DO
    myStr(tmp3) = myStr(tmp4)
    tmp3 = tmp3 + 1 /Move Hi 8 to front
    tmp4 = tmp4 + 1
    LOOP UNTIL (tmp3 > 7)
  myStr(8) = 0
  SEROUT Dsp1, 84, [STR myStr]
  EXIT
ENDIF
NEXT Go through string
IF DspStr = 0 THEN
  DEBUG DEC tmp1, ". ", STR myStr, CR
  ENDIF
DspStr = 0
NEXT Read in strings
HIGH epCs 'Set Chip Select High - Close Command
epAddr = myStrS * 16 /Reset Address
GOTO CmdDisplay

```

- Display a long String

Fun

This document was last generated on 2013-01-06 06:29:02 AM.